

Treball de Fi de Grau

**Grau en Enginyeria en Tecnologies Industrials**

**RECONeixEMEN DE TEXT MANUSCRIT  
MITJANÇANT LA SVD**

**MEMÒRIA**

**Autor:** José de la Puente Puigjaner  
**Director:** Jaume Amorós Torrent  
**Convocatòria:** Gener 2017



Escola Tècnica Superior  
d'Enginyeria Industrial de Barcelona



## Resum

Aquest treball és una investigació sobre el reconeixement de text manuscrit mitjançant la SVD (descomposició en valors singulars). És la continuació d'un projecte que es va realitzar prèviament en el qual es va investigar la implementació d'aquesta metodologia per al reconeixement de text tipografiat (Berguedà, 2016).

Té com a objectiu poder reconèixer textos manuscrits amb dos requeriments principals: les línies de text han de ser completament horitzontals i les lletres han d'estar separades per espais blancs equidistants, independentment de si pertanyen o no a la mateixa paraula.

El sistema de reconeixement es basa en l'OCR (de l'anglès *Optical Character Recognition*; és a dir, Reconeixement Òptic de Caràcters). Aquest mètode consisteix en detectar una per una les lletres del text, obtenir les components de cadascuna i comparar-les amb uns patrons. Actualment existeixen molts sistemes de reconeixement de text d'aquest tipus. La innovació d'aquest treball es basa en utilitzar la SVD per a la obtenció de components i, per tant, definir un nou sistema de comparació amb patrons.

Cada lletra detectada es transforma en una matriu de 1's i 0's (on els 1's són els píxels negres i els 0's són els blancs) i se li aplica la SVD, obtenint d'aquesta forma els valors i vectors singulars de sortida i d'arribada de cada lletra. D'aquestes components s'agafen només els primers valors i vectors, ja que són els que aporten la informació més important. En el cas dels valors singulars s'agafen els 2 primers, i en el cas dels vectors, els 3 primers. Degut a que cada valor o vector singular no aporta la mateixa informació, es dona més importància al primer valor i vector i menys als restants.

Els continguts d'aquest projecte comencen amb una introducció al reconeixement de text i una base teòrica on s'expliquen les principals propietats de la SVD que s'apliquen en aquesta metodologia, segueixen amb una explicació detallada de l'algoritme de reconeixement i posteriorment s'explica la seva implementació del codi amb programa MATLAB. Finalment es realitzen una sèrie d'estudis i es presenten els resultats.

Els estudis consisteixen en reconèixer una sèrie de textos manuscrits. El percentatge d'encerts d'aquests textos oscil·la entre un 87% i un 97%. Aquests resultats són excepcionals tenint en compte que en un principi no s'esperava superar el 20%. S'acaba conclouent que, degut a l'èxit dels resultats, és recomanable continuar investigant aquesta metodologia de la SVD per a reconèixer text manuscrit.



# Sumari

<b>RESUM .....</b>	<b>2</b>
<b>SUMARI .....</b>	<b>4</b>
<b>1. GLOSSARI .....</b>	<b>6</b>
<b>2. PREFACI.....</b>	<b>7</b>
2.1. ORIGEN DEL PROJECTE.....	7
2.2. MOTIVACIÓ .....	7
2.3. REQUERIMENTS PREVIS .....	7
<b>3. INTRODUCCIÓ .....</b>	<b>9</b>
3.1. OBJECTIUS DEL PROJECTE .....	9
3.2. ABAST DEL PROJECTE .....	9
<b>4. INTRODUCCIÓ AL RECONeixEMENT DE CARÀCTERS.....</b>	<b>10</b>
4.1. DEFINICIÓ .....	10
4.2. ESTAT DE L'ART .....	10
4.3. FUNCIONAMENT DEL SISTEMA OCR.....	12
4.3.1. Binarització.....	12
4.3.2. Segmentació de la imatge.....	12
4.3.3. Simplificació de les components.....	12
4.3.4. Comparació amb patrons.....	12
4.3.5. Problemes associats .....	13
<b>5. FONAMENTS TEÒRICS.....</b>	<b>14</b>
5.1. DEFINICIÓ PRÈVIA DE LA SVD .....	14
5.2. INVARIANTS DE LA IMATGE .....	15
5.2.1. Teorema.....	16
5.2.2. Demostració .....	17
5.2.3. Corol·lari .....	18
5.3. EXPANSIÓ DE LA SVD.....	19
5.3.1. Importància dels valors singulars.....	20
5.3.2. Importància dels vectors singulars.....	21
<b>6. ALGORITME.....</b>	<b>25</b>
6.1. REQUISITS DE LA IMATGE .....	25
6.2. PAS PREVI .....	25
6.3. BINARITZACIÓ DE LA IMATGE .....	26
6.4. SEGMENTACIÓ PER LLETRES.....	26
6.5. OBTENCIÓ DE COMPONENTS .....	27
6.6. COMPARACIÓ AMB EL CATÀLEG.....	28

6.6.1.	<i>Creació del catàleg</i> .....	28
6.6.2.	<i>Procediment de comparació</i> .....	28
6.7.	DETERMINACIÓ DE L'ÈXIT DEL REONEIXEMENT .....	31
<b>7.</b>	<b>IMPLEMENTACIÓ DEL CODI</b> .....	<b>32</b>
7.1.	FUNCIONS PRINCIPALS.....	32
7.2.	FUNCIONS SECUNDÀRIES .....	35
7.3.	CONTROL DE LES FUNCIONS.....	36
7.4.	CREACIÓ DELS CATÀLEGS.....	37
7.5.	FUNCIONS DE FLUX DE TINTA .....	37
7.6.	FUNCIONS AUXILIARS .....	38
7.7.	GESTIÓ DOCUMENTAL.....	39
<b>8.</b>	<b>ESTUDIS REALITZATS</b> .....	<b>40</b>
8.1.	CATÀLEGS .....	40
8.2.	TEXTOS A ESTUDIAR .....	41
8.3.	DETERMINACIÓ DEL MODEL DEL CATÀLEG .....	41
8.4.	DETERMINACIÓ DELS PESOS DELS VALORS I VECTORS SINGULARS.....	43
8.5.	ANÀLISI D'ERRORS TÍPICS .....	46
8.5.1.	<i>Discriminació per tipus de caràcter</i> .....	46
8.5.2.	<i>Parelles de caràcters</i> .....	46
8.6.	ESTUDIS AMB EL SISTEMA FINAL .....	48
<b>9.</b>	<b>PLANIFICACIÓ I PROGRAMACIÓ</b> .....	<b>50</b>
<b>10.</b>	<b>PRESSUPOST</b> .....	<b>51</b>
<b>11.</b>	<b>IMPACTE SOBRE L'ENTORN</b> .....	<b>52</b>
11.1.	IMPACTE MEDIAMBIENTAL .....	52
11.2.	IMPACTE SOCIAL.....	52
	<b>CONCLUSIONS</b> .....	<b>53</b>
	<b>AGRAÏMENTS</b> .....	<b>55</b>
	<b>BIBLIOGRAFIA</b> .....	<b>56</b>

# 1. Glossari

**Estructura:** Terme de MATLAB que es refereix a una estructura d'emmagatzematge d'informació per cel·les.

**OCR:** De l'anglès, *Optical Character Recognition* (Reconeixement Òptic de Caràcters).

**String:** Terme de MATLAB que es refereix a una cadena de caràcters.

**SVD:** De l'anglès, *Singular Value Decomposition* (Descomposició en Valors Singulars).

## 2. Prefaci

### 2.1. Origen del projecte

En els últims anys la tecnologia de reconeixement de text s'ha anat donant a conèixer cada cop més i s'ha desenvolupat obtenint cada vegada millors resultats. Encara així, cap de les metodologies de reconeixement de text més conegudes es basa en la SVD.

Partint d'això, al Juny de 2016 la Judith Berguedà va presentar el projecte *Reconeixement de text mitjançant la SVD* (Berguedà, 2016), amb Jaume Amorós com a director de projecte, en el qual es va adaptar el sistema OCR per basar-lo en la SVD. En aquest projecte es van realitzar proves de text tipografiat amb 3 alfabetes diferents (llatí, grec i rus) i els resultats obtinguts van ser molt bons.

Aquest és una continuació del projecte de la Judith Berguedà, en el qual es vol continuar investigant la metodologia de la SVD per al reconeixement de text manuscrit.

### 2.2. Motivació

La motivació principal d'aquest projecte és continuar amb la investigació iniciada en el projecte previ, portant el reconeixement de text mitjançant la SVD al següent nivell. Com ja s'indica el títol, el següent nivell es poder reconèixer no només text tipografiat, sinó també text manuscrit.

Com ja s'ha comentat, avui en dia existeixen sistemes de reconeixement de text tipografiat, però no hi ha tants que puguin reconèixer text manuscrit amb una precisió molt elevada. Degut a l'èxit de la implementació de la SVD en el projecte previ, es vol estudiar si aquest sistema serà igualment efectiu amb text manuscrit.

### 2.3. Requeriments previs

Els requeriments previs més importants per a la realització d'aquest projecte són els coneixements d'àlgebra lineal i de programació amb MALAB.

Els coneixements d'àlgebra lineal són necessaris per al desenvolupament de l'algoritme de reconeixement de text. Concretament, els coneixements de la descomposició en valors singulars i les seves propietats són trivials, ja que aquest projecte es basa principalment en la SVD.

MATLAB és la eina que proporciona la implementació pràctica de l'algoritme a casos reals, sense la qual no es podrien obtenir resultats experimentals. Per tant, és necessari tenir un

coneixement avançat de programació en aquest programa.

S'ha escollit MATLAB degut a que és un programa que presenta moltes facilitats en un entorn de proves. També ha afectat el fet de disposar d'una llicència d'estudiant gratuïta proporcionada per la UPC. Encara així, tot el codi desenvolupat es pot fer servir en el programa Octave, de llicència lliure.

Per últim, la informació aportada per el projecte previ a aquest (Berguedà, 2016) és també molt important per poder partir d'una base de coneixement per a realitzar aquest projecte.



## 3. Introducció

### 3.1. Objectius del projecte

Partint del projecte previ (Berguedà, 2016) on es va implementar la SVD en la metodologia OCR per a reconeixement de text manuscrit, l'objectiu principal d'aquest projecte és arribar a reconèixer text manuscrit implementant les adaptacions i millores necessàries.

Per tal de complir aquest propòsit, es volen tractar els següents temes:

- Implementar un nou algoritme de reconeixement de text manuscrit que doni més importància als vectors singulars.
- Estudiar com afecta en la precisió de reconeixement la informació aportada per els valors i vectors singulars i aplicar aquests coneixements a l'algoritme per tal d'obtenir la màxima precisió.
- Identificar els errors principals de reconeixement.

### 3.2. Abast del projecte

Degut a que el reconeixement de text manuscrit amb la SVD es troba en la seva fase inicial no es considerarà l'estudi de paraules senceres ni de símbols en aquest projecte. Per concretar, les característiques dels textos a analitzar seran les següents:

- Text escrit en línies perfectament horitzontals.
- Lletres separades per espais en blanc de forma equidistant (independentment de si pertanyen a la mateixa paraula o no).
- No es detectaran accents, per tant s'utilitzarà l'alfabet anglès.
- No es detectarà cap tipus de símbol (signes de puntuació, exclamació...)

Tot i que les característiques indicades són molt restrictives, es pot crear i utilitzar qualsevol tipus de catàleg que contingui símbols, signes de puntuació, altres alfabetes... Per tant, la utilitat de l'algoritme no es limita únicament als estudis realitzats en aquest projecte.

## 4. Introducció al reconeixement de caràcters

En el món actual hi ha una creixent tendència de digitalitzar-ho tot. En els últims anys, el reconeixement de caràcters ha començat a destacar notablement. Uns exemples clars d'això són: impressores que reconeixen el text d'una targeta d'identificació, cotxes que detecten la velocitat de la carretera a partir d'un senyal vertical, reconeixement de matricules de vehicles, traductors que tradueixen el text instantàniament a través de la càmera d'un smartphone... I com aquests, milers d'exemples més.

Les possibilitats que ofereix aquest sistema creixen conforme avança aquesta tecnologia. Per tant, és interessant estudiar-la.

### 4.1. Definició

El reconeixement òptic de text, generalment reconegut com reconeixement de caràcters i expressat amb freqüència amb les sigles OCR (de l'anglès *Optical Character Recognition*), és un procés dirigit a la digitalització de textos, els quals identifiquen automàticament a partir d'una imatge, símbols o caràcters que pertanyen a un determinat alfabet, per després guardar-los en forma de dades (Reconocimiento Óptico de Caracteres, 2016).

### 4.2. Estat de l'art

Actualment existeixen diverses metodologies de reconeixement de text. Algunes de les més conegudes avui en dia són les següents (Worldpress, 2009):

- **OMR** (*Optical Mark Recognition*): És el més simple dels sistemes. Tal com reflexa el seu nom, l'OMR no detecta caràcters, sinó marques òptiques. En concret, detecta la presència o absència de les mateixes. Per que funcioni correctament és necessari marcar els camps amb un color opac. A més, es necessiten unes marques de referència al full. Si es compleixen les condicions, aquest sistema arriba fins a un 99,9% de precisió. S'utilitza per la correcció automàtica d'exàmens tipus test, formularis, enquestes i altres documents de característiques similars. A la Figura 4.1 es mostra un full de marques òptiques.

[illegible]

*Figura 4.1 Full de marques òptiques per al reconeixement amb OMR. Imatge (Worldpress, 2009)*

### **4.3. Funcionament del sistema OCR**

El mètode OCR consta de quatre fases fonamentals: binarització, segmentació, simplificació de components i comparació amb els patrons.

#### **4.3.1. Binarització**

El primer pas té com a objectiu distingir el text dintre de la imatge. Partint d'una imatge en color o en escala de grisos és convenient convertir-la a una imatge en blanc i negre de tal forma que es preservin les propietats de la imatge. Amb aquest propòsit, s'escull un llindar adequat a partir del qual tots els píxels que no el superin es convertiran en negre i la resta en blanc.

Amb aquest procés s'obté una imatge en blanc i negre on queden clarament marcats els contorns dels caràcters que conté la imatge.

#### **4.3.2. Segmentació de la imatge**

Aquest es el procés més costós i necessari per al posterior reconeixement del text. Com el propi nom indica, la segmentació de la imatge consisteix en fraccionar la imatge per separar els diferents caràcters que conté.

No existeix un mètode genèric per a realitzar aquesta segmentació. Dues de les tècniques més utilitzades són les projeccions lineals i la determinació dels agrupaments (clústers) de píxels mitjançant un histograma.

#### **4.3.3. Simplificació de les components**

Una vegada aïllats els components de la imatge, serà necessari aplicat un procés de simplificació dels components per a cadascun d'ells. Aquest procediment consisteix en eliminar successivament els punts dels contorns de cada component de forma que es conservi la seva tipologia.

En el cas d'aquest projecte aquesta fase és completament diferent, ja que la obtenció de components de la imatge es basa en la SVD.

#### **4.3.4. Comparació amb patrons**

En aquesta última fase es comparen els caràcters obtinguts amb uns patrons emmagatzemats en una base de dades. Aquesta etapa es la més determinant en quant a la precisió.

Un dels mètodes principals per dur a terme aquesta comparació és el de les projeccions. Encara així, cap dels mètodes habituals serviran per a realitzar aquesta comparació (degut a la SVD), pel que serà necessari desenvolupar un nou sistema amb aquest propòsit.

#### **4.3.5. Problemes associats**

El principal problema en qualsevol sistema de reconeixement de text són les imperfeccions. Aquestes poden ser causades durant l'escaneig de la imatge o poden ser taques que ja hi eren al full. Sigui quin sigui l'origen, aquestes causen errors de detecció al no poder diferenciar les imperfeccions del que és realment una lletra. Aquest problema també estarà molt present en el desenvolupament del projecte.

## 5. Fonaments teòrics

### 5.1. Definició prèvia de la SVD

(Amorós, 2010) i (Descomposició en valores singulares, 2016)

#### Definició

Sigui  $A$  una matriu real qualsevol de mida  $m \times n$ . Aleshores, la matriu  $S = A^t A$  quadrada, és simètrica i semidefinida positiva. Per tant els seus valors propis són sempre reals i majors o iguals a 0.

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0 \quad (5.1)$$

Veiem que:

$$A^t A = A^t (A^t)^t = (A^t A)^t \quad (5.2)$$

Per tant,  $S$  és simètrica.

També veiem que:

$$(Ax, Ax) = A^t A^t A x = \|Ax\|^2 \geq 0 \quad (5.3)$$

Per tant,  $S = A^t A$  és semidefinida positiva.

#### Definició

Siguin  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0$  els autovalors no nuls de la matriu  $S = A^t A$  ordenats de major a menor. Llavors  $\sigma_i = \sqrt{\lambda_i}$  és el  $i$ -èssim valor singular de la matriu  $A$ .

#### Teorema

Sigui  $A$  la matriu d'una aplicació lineal  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$  amb autovalors  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r \geq 0$  de la matriu  $S = A^t A$ . Llavors existeix una descomposició  $A = U \cdot D \cdot V^t$  en bases canòniques de sortida i d'arribada, amb  $U$  de mida  $m \times m$  ortogonal,  $V$  de mida  $n \times n$  ortogonal i

$$D = \begin{pmatrix} \sigma_1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & \sigma_r & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 \end{pmatrix} \quad (5.4)$$

tals que  $D$  és una matriu de  $f$  en bases ortonormals  $v_1, \dots, v_n$  de sortida i  $u_1, \dots, u_m$

d'arribada.

Llavors:

1.  $(Av_1, \dots, Av_r)$  és el conjunt ortogonal i  $\|Av_i\| = \sigma_i = \sqrt{\lambda_i}$
2.  $\left(\frac{Av_1}{\sigma_1}, \dots, \frac{Av_r}{\sigma_r}\right)$  és una base ortonormal del subespai fonamental  $Col(A)$ .
3.  $(v_{r+1}, \dots, v_n)$  és una base ortonormal del subespai fonamental  $Nul(A)$ .
4.  $rang(A) = r$  coincideix amb el nombre de valors singulars no nuls.

Així, la descomposició  $A = U \cdot D \cdot V^t$  és, per tant, el canvi de base

$$M_e^e(f) = C_e^u M_e^u(f) C_v^e \quad (5.5)$$

### Demostració

1.  $(Av_i, Av_j) = v_i^T A^T Av_j = \lambda_j \cdot v_i^T v_j = \begin{cases} \lambda_j & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$ . Tenint en compte el resultat  $\|Av_i\| = \sqrt{(Av_i, Av_i)} = \sqrt{\lambda_i} = \sigma_i$
2. Com el conjunt de vectors  $v_i, 1 \leq i \leq r$  és ortonormal (per tant linealment dependent), el producte  $Av_i$  és una combinació lineal de les columnes de la matriu. Per tant, tenint en compte el que s'ha demostrat a (1),  $\left(\frac{Av_1}{\sigma_1}, \dots, \frac{Av_r}{\sigma_r}\right)$  és una base ortonormal de  $Col(A)$ .
3. Tenint  $v_i, r+1 \leq i \leq n$  vectors associats als autovalors nuls, tenint en compte el que s'ha demostrat a (1) i considerant que  $Nul(A) = Nul(A^t A)$ , es veu que  $(v_{r+1}, \dots, v_n)$  és una base ortonormal de  $Nul(A)$ .
4. Fixant-nos en la dimensió del subespai trobat en (2), queda clar que  $rang(A) = r$ .

## 5.2. Invariants de la imatge

Per obtenir la informació d'un caràcter, aquest es transforma en una matriu  $M$  on s'assigna valor 1 als píxels negres i valor 0 als blancs (com podem veure a la Figura 5.1) i es calcula la SVD.



Figura 5.1 Imatge de píxels blancs i negres

Un cop realitzat això, es presenta un problema. Degut a que el caràcter pot ser escanejat amb diferents resolucions, s'obtenen matrius diferents, de forma que l'escala pot variar fins i tot de forma diferent als eixos X,Y.



Figura 5.2 Variacions del mateix caràcter degut als canvis d'escala dels eixos X,Y

Per tal que això no afecti als resultats, es necessari extreure les propietats de la SVD de la matriu de forma que siguin invariants a aquests canvis d'escala.

### 5.2.1. Teorema

Considerarem separatament els canvis d'escala en els eixos OX, OY.

#### 1. Canvis d'escala en l'eix OX (repetició de columnes)

Donada una matriu  $M$  d'un caràcter qualsevol, obtenim una nova matriu  $\tilde{M}$  canviant cada coeficient de  $M$  per una fila amb  $n$  coeficients iguals:

$$M = \begin{pmatrix} 0 & 0 & \dots \\ 0 & 1 & \dots \\ 1 & \vdots & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \longrightarrow \tilde{M} = \begin{pmatrix} \overbrace{0 \dots 0}^n & \overbrace{0 \dots 0}^n & \dots \\ \overbrace{0 \dots 0}^n & \overbrace{1 \dots 1}^n & \dots \\ \overbrace{1 \dots 1}^n & \vdots & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad (5.6)$$

La relació entre les SVD de  $M$  i de  $\tilde{M}$  és:

- Els valors singulars no nuls de  $\tilde{M}$  són els de  $M$ , multiplicats per  $\sqrt{n}$
- Si  $\sigma \neq 0$  és un valor singular de  $M$ , amb vector singular de sortida  $v = (v_1, v_2, \dots)$  el valor singular  $\tilde{\sigma} = \sigma \cdot \sqrt{n}$  de  $\tilde{M}$  té vector singular de sortida  $\tilde{v} = \left( \overbrace{\frac{v_1}{\sqrt{n}}, \dots, \frac{v_1}{\sqrt{n}}}^n, \overbrace{\frac{v_2}{\sqrt{n}}, \dots, \frac{v_2}{\sqrt{n}}}^n, \dots \right)$ . Es a dir, cada coeficient de  $v$  es repeteix  $n$  cops dividit per  $\sqrt{n}$ .
- Si  $\sigma \neq 0$  és un valor singular de  $M$ , el vector singular d'arribada de  $\tilde{\sigma} = \sigma \cdot \sqrt{n}$  de  $\tilde{M}$  és el mateix vector singular d'arribada que el de  $\sigma$  de  $M$ .



## 2. Canvis d'escala en l'eix OY (repetició de files)

Donada una matriu  $M$  d'un caràcter qualsevol, obtenim una nova matriu  $\hat{M}$  canviant cada coeficient de  $M$  per una columna amb  $m$  coeficients iguals:

$$M = \begin{pmatrix} 0 & 0 & \dots \\ 0 & 1 & \dots \\ 1 & \vdots & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \longrightarrow \hat{M} = \begin{pmatrix} m \begin{Bmatrix} 0 \\ \vdots \\ 0 \end{Bmatrix} & m \begin{Bmatrix} 0 \\ \vdots \\ 1 \end{Bmatrix} & \dots \\ m \begin{Bmatrix} 0 \\ \vdots \\ 0 \end{Bmatrix} & m \begin{Bmatrix} 1 \\ \vdots \\ 1 \end{Bmatrix} & \dots \\ m \begin{Bmatrix} 1 \\ \vdots \\ 1 \end{Bmatrix} & \vdots & \ddots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad (5.7)$$

La relació entre les SVD de  $M$  i de  $\hat{M}$  és:

- Els valors singulars no nuls de  $\hat{M}$  són els de  $M$ , multiplicats per  $\sqrt{m}$
- Si  $\sigma \neq 0$  és un valor singular de  $M$ , el vector singular de sortida de  $\tilde{\sigma} = \sigma \cdot \sqrt{m}$  de  $\hat{M}$  és el mateix vector singular d'arribada que el de  $\sigma$  de  $M$ .
- Si  $\sigma \neq 0$  és un valor singular de  $M$ , amb vector singular de d'arribada  $u = (u_1, u_2, \dots)$  el valor singular  $\tilde{\sigma} = \sigma \cdot \sqrt{m}$  de  $\hat{M}$  té vector singular de sortida  $\tilde{u} = \left( \overbrace{\frac{u_1}{\sqrt{m}}, \dots, \frac{u_1}{\sqrt{m}}}^m, \overbrace{\frac{u_2}{\sqrt{m}}, \dots, \frac{u_2}{\sqrt{m}}}^m, \dots \right)$ . Es a dir, cada coeficient de  $u$  es repeteix  $m$  cops dividit per  $\sqrt{m}$ .

### 5.2.2. Demostració

Les matrius  $\tilde{M}$  i  $\hat{M}$  són productes tensorials de Kronecker:

$$\tilde{M} = M \otimes \underbrace{(1 \ 1 \ \dots \ 1)}_n, \quad \hat{M} = M \otimes \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1_m \end{pmatrix} \quad (5.8)$$

de manera que el teorema és corol·lari del teorema sobre la SVD d'un producte tensorial (Horn & Johnson, 1991).

D'aquí s'obtenen els invariants que s'estan buscant.

### 5.2.3. Corol·lari

Sigui  $M$  la matriu numèrica d'un caràcter, amb valors singulars  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$  vectors singulars de sortida  $v_1 \geq v_2 \geq \dots \geq v_n$  i vectors singulars d'arribada  $u_1 \geq u_2 \geq \dots \geq u_m$ . Aleshores:

- Els quocients pel valor singular principal  $\frac{\sigma_1}{\sigma_1}, \frac{\sigma_2}{\sigma_1}, \dots, \frac{\sigma_r}{\sigma_1}$  no varien si es fa qualsevol reescalat dels eixos OX, OY.
- Considerem cada vector singular de sortida o d'arribada de  $M$  com una taula de valors d'una funció  $f_u: [0,1] \rightarrow \mathbb{R}$  que subdivideix l'interval  $[0,1]$  en  $d$  parts iguals  $\left[0, \frac{1}{d}\right], \left[\frac{1}{d}, \frac{2}{d}\right], \dots, \left[\frac{d-1}{d}, 1\right]$  (on  $d$  és la longitud del vector), i té per valor constant cada component del vector  $u$  en el seu subinterval.

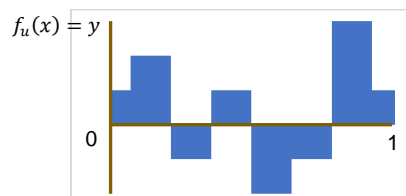


Figura 5.3 Funció subdividida en  $d$  intervals entre 0 i 1

Si es normalitza aquesta funció (Figura 5.3) per a que tingui norma  $L^2$  igual a 1 en l'interval  $[0,1]$ , la funció normalitzada no varia si es fa qualsevol reescalat dels eixos OX,OY.

Veiem que (a) és conseqüència immediata del teorema i (b) prové de calcular la norma  $L^2$  de les funcions  $f_u$  o  $f_v$ . Aleshores, es veu que cada reescalat en l'eix OX o OY divideix tota la funció per  $\sqrt{n}$  o  $\sqrt{m}$  respectivament.

Degut a aquestes propietats d'invariància per canvi d'escala, les components que cal estudiar per a identificar els caràcters a partir de la seva matriu de 1's i 0's és:

- Quocients resultants de dividir cada valor singular per el primer d'ells:  $\frac{\sigma_1}{\sigma_1}, \frac{\sigma_2}{\sigma_1}, \dots, \frac{\sigma_r}{\sigma_1}$
- Vectors singulars de sortida  $[v_1, v_2, \dots, v_n]$  i d'arribada  $u_1, u_2, \dots, u_m$ , considerats com funcions de  $[0,1]$  en  $\mathbb{R}$ .

Tenint en compte que el quocient del primer valor singular  $\frac{\sigma_1}{\sigma_1}$  sempre resulta 1 i no aporta cap valor extra, no es tindrà en compte en tot l'estudi. Per tant, a partir d'aquest punt, sempre que es nombrin els primers valors singulars es farà referència al quocients dels valors singulars sense comptar el primer:  $Valors\ singulars = \frac{\sigma_2}{\sigma_1}, \frac{\sigma_3}{\sigma_1}, \dots, \frac{\sigma_r}{\sigma_1}$

### 5.3. Expansió de la SVD

**Lema 1** (Berguedà, 2016)

Mitjançant la SVD, qualsevol matriu real  $A$  de rang  $r$  es pot escriure com:

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T \quad (5.9)$$

on  $u_i, \sigma_i, v_i$  són el triplet i-èssim singular d' $A$ .

Demostració:

$$A = U \cdot D \cdot V^t = [u_1, \dots, u_r, \dots, u_n] \begin{bmatrix} \sigma_1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & \sigma_r & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \\ \vdots \\ v_k^T \end{bmatrix}; n, k \geq r \quad (5.10)$$

$$\Rightarrow A = [u_1, \dots, u_r] \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_r \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_r \end{bmatrix} = [\sigma_1 u_1, \dots, \sigma_r u_r] \begin{bmatrix} v_1 \\ \vdots \\ v_r \end{bmatrix} = \quad (5.11)$$

$$= \sigma_1 u_1 v_1^T, \dots, \sigma_r u_r v_r^T \quad (5.12)$$

$$\Rightarrow A = \sum_{i=1}^r \sigma_i u_i v_i^T \quad (5.13)$$

La representació  $A = U \cdot D \cdot V^t$  s'anomena expansió SVD. Amb aquests components es pot reconstruir la matriu  $A$ . De tots aquests termes, els últims (aproximadament els últims  $r - k$  termes) aporten un valor negligible, ja que:

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0 \quad (5.14)$$

Per tant, utilitzant únicament els primers valors singulars (els més significatius), la matriu  $A$  es pot aproximar com:

$$A \approx \sum_{i=1}^r \sigma_i u_i v_i^T, k < r \quad (5.15)$$

### 5.3.1. Importància dels valors singulars

A la Figura 5.4 podem veure com els primers valors són els que aporten més informació i conforme s'agafen més la informació aportada decreix.

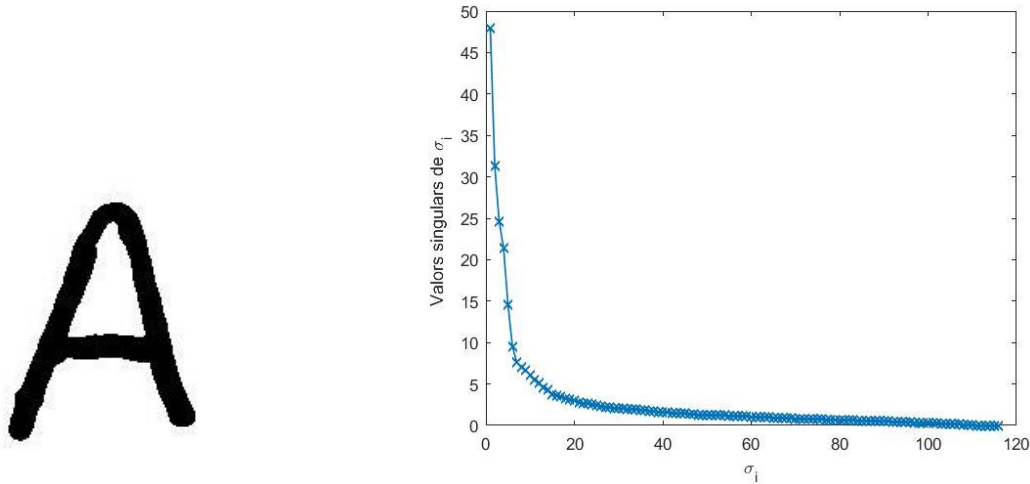


Figura 5.4 Valors singulars obtinguts de la lletra A

En la Figura 5.5 podem observar com queda reconstruïda la lletra A en funció dels valors singulars que s'utilitzen.



Figura 5.5 Lletres A reconstruïda a partir de 1, 2, 3, 4 i 5 valors singulars

La informació aportada per els valors menys rellevants acostumen a ser parts de la imatge que no pertanyen a la lletra (soroll de la imatge), com pot ser un píxel que no pertany al caràcter, causat per una taca o una imperfecció durant l'escaneig. Aplicant aquest concepte al projecte, es pot assegurar que els últims valors singulars no només no aporten informació sinó que causen errors a l'hora de detectar una lletra. Per tant, s'ha de definir quin nombre de valors singulars s'utilitzaran per el reconeixement.

A la Figura 5.6 podem veure representats gràficament els primers valors singulars de 6 mostres de la lletra A i 6 mostres de la lletra B.

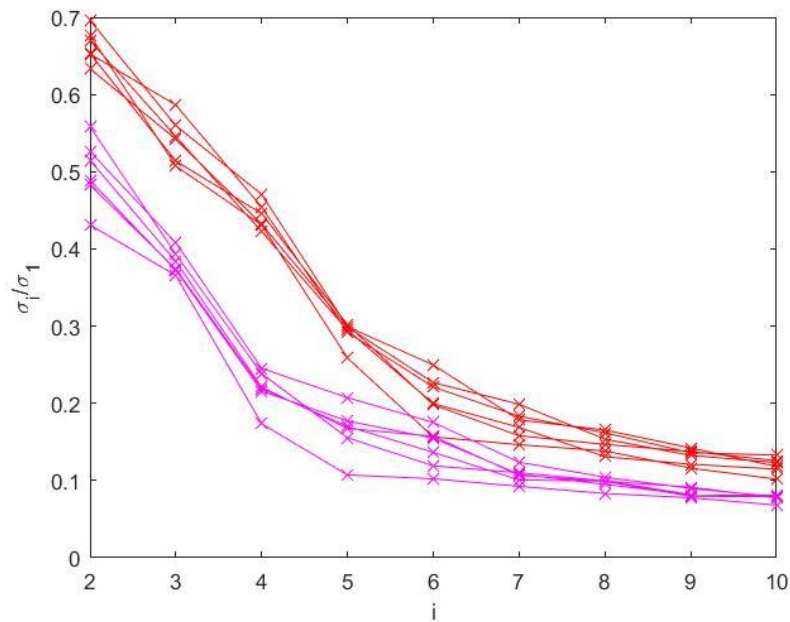


Figura 5.6 Representació dels valors singulars d'A (color vermell) i de B (color magenta)

Com podem observar, en els valors 2, 3 i 4 hi ha una diferència notable entre les dos lletres. A partir del 5è valor singular aquesta diferència cada vegada és més curta. Per tant, per a realitzar l'estudi es tindran en compte un màxim de 3 valors singulars:  $\left[\frac{\sigma_2}{\sigma_1}, \frac{\sigma_3}{\sigma_1}, \frac{\sigma_4}{\sigma_1}\right]$

### 5.3.2. Importància dels vectors singulars

A l'hora de reconstruir una lletra es necessita tant dels valors singulars com dels vectors singulars. Ja que, com s'ha mencionat abans:

$$A = U \cdot D \cdot V^t \quad (5.16)$$

on  $V = [v_1, \dots, v_n]$  són els vectors singulars de sortida i  $U = [u_1, \dots, u_m]$  els vectors singulars d'arribada.

Per tant, els vectors singulars també aporten una informació extra que no aporten els valors singulars. Aquesta informació permet, entre altres, diferenciar dues lletres simètriques.

Per entendre-ho millor, veurem el següent exemple (Berguedà, 2016):

En el cas de les lletres amb tipografia d'ordinador, la simetria entre algunes lletres pot arribar a ser perfecta. En la Figura 5.7 podem veure el cas de les lletres b, d, p i q.

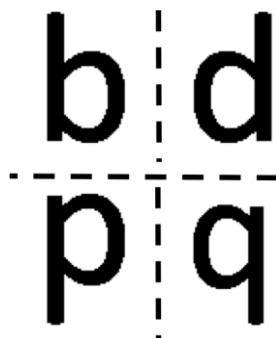


Figura 5.7 Simetries entre les lletres b, d, p i q. Imatge (Berguedà, 2016)

Tal com es mostra als gràfics de la Figura 5.8 les lletres b, d i p tenen exactament els mateixos valors singulars (amb diferències ínfimes de l'ordre de  $10^{-14}$ ). Per tant, els valors singulars no serveixen per diferenciar-les i hem de fer servir dels vectors singulars.

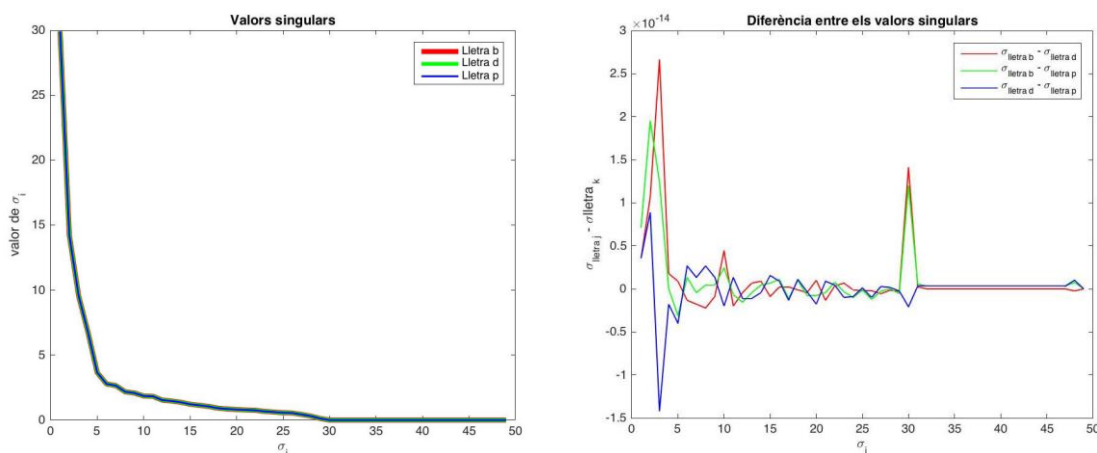


Figura 5.8 Comparació els valors singulars de b, d i p. Imatge (Berguedà, 2016)

Degut a que en aquest projecte s'analitza text manuscrit, aquestes mateixes lletres no són perfectament simètriques, tal com podem veure a la Figura 5.9.

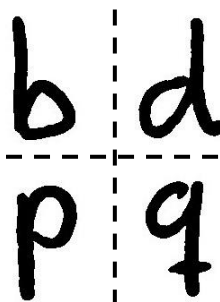


Figura 5.9 Simetries entre les lletres manuscrites b, d, p i q

En aquest cas, les lletres més semblants són la b i la p. A la Figura 5.10 observem que els 3 primers valors singulars, tot i que no tenen exactament el mateix valor són prou semblants com per causar un error de reconeixement.

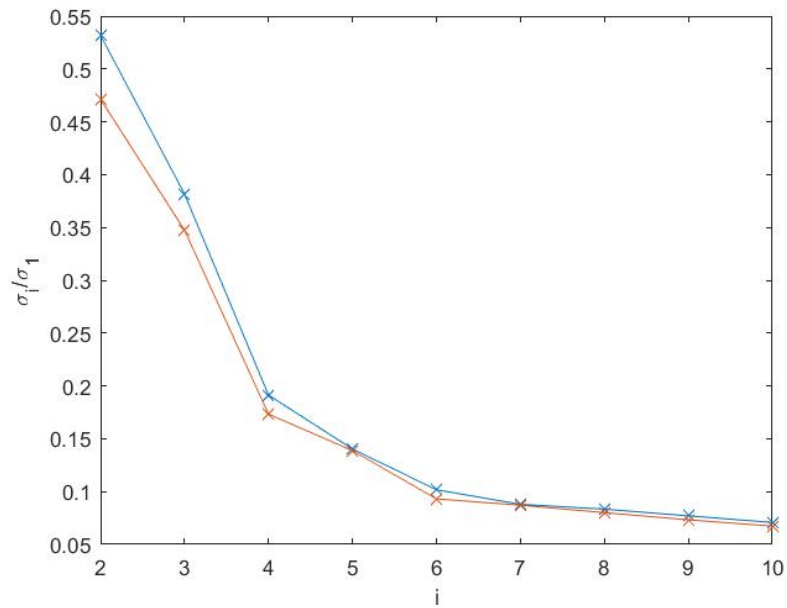


Figura 5.10 Representació dels primers valors singulars de les lletres b i p

Quocient	b	d
$\sigma_2/\sigma_1$	0.53169	0.47136
$\sigma_3/\sigma_1$	0.3816	0.34734
$\sigma_4/\sigma_1$	0.1913	0.1733

Taula 5.1 Comparació dels 3 primers valors singulars de les lletres b i d

En el cas que acabem de veure només es compara una mostra de cada lletra. Com s'ha vist a l'apartat anterior (0) els valors singulars de cada lletra varien dins d'un rang, ja que la mateixa lletra no sempre s'escriu de la mateixa forma. Per tant, els valors de b i p són tan semblants que amb una petita variació es pot cometre un error de reconeixement. A la Figura 5.11 s'ha representat un diagrama de blocs mostrant el rang dels valors singulars per a 6 mostres de b i 6 de p.

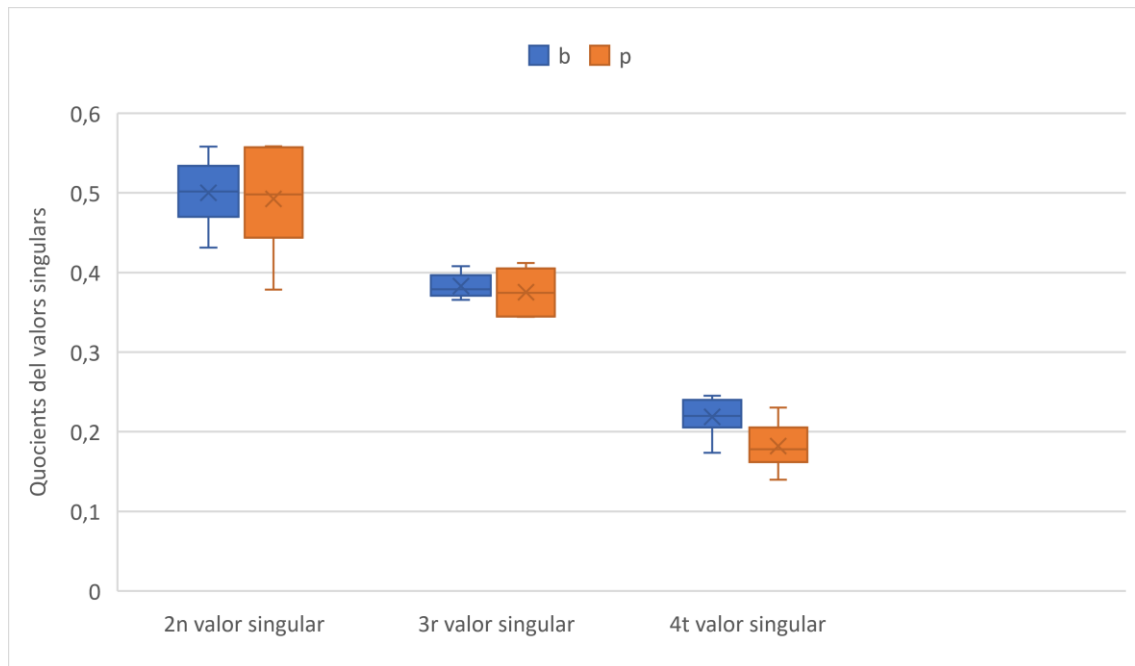


Figura 5.11 Diagrama de caixes que representen els valors singulars de 6 mostres de b i p

Observem que, efectivament, els rangs es solapen i la probabilitat de cometre un error de reconeixement és molt alta. Per tant, els vectors singulars són necessaris per diferenciar-les.

Aquesta situació en que els valors singulars de dos lletres són molt semblants no es dona només en casos de simetria. També es pot donar en qualsevol parell de lletres que tinguin unes característiques semblants (en 8.5.2 es mostren els casos més habituals). Per tant, la informació aportada per els vectors singulars és una part fonamental d'aquest projecte.



## 6. Algoritme

El reconeixement de text manuscrit consta de quatre fases: binarització de la imatge, segmentació per lletres, obtenció de components i comparació amb el catàleg.

Prèviament es necessari tractar la imatge per a que compleixi una sèrie de requeriments i corregir les imperfeccions que apareixen durant l'escaneig (taques, reflexos i marges irregulars).

### 6.1. Requisits de la imatge

Per al correcte funcionament del codi, el text a reconèixer ha d'estar escrit en línies horitzontals i les lletres han d'estar separades de forma equidistant (independentment de si pertanyen o no a la mateixa paraula). Respecte a les característiques de la imatge que conté el text, ha de complir dos condicions principals: estar en blanc i negre i en format TIFF.

A part d'això, es necessari eliminar les cantonades irregulars amb qualsevol programa d'edició de imatge.

### 6.2. Pas previ

Els reflexos causats durant l'escaneig provoquen que alguns píxels continguts en un caràcter siguin blancs quan haurien de ser negres, tal com podem veure a la Figura 6.1. Això afecta molt negativament a l'hora de reconèixer la lletra, ja que fa variar molt els valors i vectors singulars. Per tal de solucionar-ho, la imatge es tracta amb un flux de tinta negra.



*Figura 6.1 Lletra A escanejada amb reflexos*

El nom “flux de tinta negra” fa referència a com quedaria la imatge si els caràcters estiguessin escrits amb tinta líquida i deixéssim que aquesta s'escampés per el full. Com

s'explica més detalladament en el següent apartat (6.3), cada píxel té un valor d'1 o 0 en funció de si és blanc o negre. El procediment consisteix en recórrer la imatge (que realment és una matriu de 1's i 0's) píxel per píxel realitzant la següent operació:

1. La tinta continguda en el píxel queda reduïda a un 60%; es a dir, es multiplica el valor que contingui aquest píxel (entre 0 i 1) per 0,6.
2. Al seu torn, els píxels laterals, superior i inferior reben un 12,5% de la tinta del píxel en qüestió; es a dir, se'ls hi suma el valor d'aquest píxel multiplicat per 0,125.

Després de realitzar aquesta iteració 9 cops, la imatge queda en una escala de grisos, ja que al realitzar aquestes operacions amb cada punt de la matriu aquests ja no tenen un valor exacte de 1's i 0's. Per tant, s'arrodoneixen a la unitat obtenint de nou una imatge purament en blanc i negre.

Amb aquest tractament els caràcters queden reforçats i les petites taques s'escampen i desapareixen. A la Figura 6.2 podem veure un exemple de l'abans i el després de realitzar aquest procés.



*Figura 6.2 Resultat d'aplicar un flux de tinta a la lletra A*

### 6.3. Binarització de la imatge

Per poder treballar amb la imatge, aquesta es transforma en una matriu de 1's i 0's on els 1 són els píxels negres (es a dir, on hi ha text escrit) i 0 els blancs. Per tant, la imatge queda com una matriu de dimensions iguals al nombre de píxels d'alçada i d'amplada de la imatge.

### 6.4. Segmentació per lletres

Partint de la matriu obtinguda a la binarització, es necessari localitzar les lletres dins de la mateixa.

Primer de tot es recorre la matriu detectant les franges horitzontals on hi ha 0's consecutius. D'aquesta forma es defineixen els límits superior i inferior de les línies del text. Encara que el tractament de flux de tinta (al pas previ) elimina algunes taques, encara en queden algunes. Per això, les línies que són significativament més petites que l'alçada mitjana de les línies de tot el text queden eliminades, ja que es pot assegurar amb certesa que el que s'ha detectat és una taca.

Una vegada queden definides les línies, es recorre cadascuna d'elles per detectar les lletres que contenen. Amb el mateix procediment utilitzat abans, es detecten les franges verticals de 0's consecutius delimitant així les lletres de cada línia. De la mateixa forma s'eliminen les taques suprimint les lletres que tenen una amplada anormalment petita en comparació amb l'amplada mitjana de la resta de lletres de la mateixa línia.

Degut a que les lletres tenen alçades diferents, es erroni que les aquestes quedin delimitades superior i inferiorment per el límit comú impostat per línia (com es pot veure a la Figura 6.3). Per tant, un últim pas detecta aquests límits i els ajusta per a cadascuna de les lletres.



Figura 6.3 Límits de la línia

Finalment, la posició de cada lletra dins de la matriu principal queda definida per quatre punts: les seves cantonades.



Figura 6.4 Delimitació de la lletra A

## 6.5. Obtenció de components

En aquesta fase s'extreuen les components principals de cada lletra; es a dir, els seus valors i vectors singulars.

Cada lletra és una matriu de 1's i 0's extreta de la matriu principal de tota la imatge. Aplicant la SVD a aquesta matriu s'obtenen els valors singulars i els vectors singulars de sortida i d'arribada. Seguidament, els valors singulars queden dividits per el valor del primer valor singular.

Finalment, queden registrats com a components de la lletra els següents valors:

- els valors singulars  $\left[\frac{\sigma_2}{\sigma_1}, \frac{\sigma_3}{\sigma_1}\right]$ ,
- els 3 primers vectors singulars de sortida  $[u_1, u_2, u_3]$
- i els 3 primers vectors singulars d'arribada  $[v_1, v_2, v_3]$ .

Com s'explica a 5.3 i es calcula experimentalment a 8.4, aquests son els valors necessaris i suficients per al reconeixement de text.

## 6.6. Comparació amb el catàleg

### 6.6.1. Creació del catàleg

El catàleg està compost per lletres majúscules, minúscules i nombres. Degut a que una lletra no s'escriu sempre igual, cadascuna d'aquestes lletres es troba repetida deu cops. Cadascuna d'aquestes repeticions estan escrites en diferents condicions (posició de la lletra en el full, velocitat d'escriptura, fatiga...). A 8.3 es demostra experimentalment la millora dels resultats d'un catàleg amb les lletres repetides enfront d'un catàleg amb una sola repetició per a cada lletra.

Aquest catàleg es realitza a partir d'una plantilla amb les lletres manuscrites i un fitxer de text amb les lletres reals assignades. Realitzant les tres primeres fases de l'algoritme, s'obtenen les components de cada lletra i es guarden en una base de dades.

### 6.6.2. Procediment de comparació

Per reconèixer una lletra es busca en catàleg la que tingui les components més semblants. Iterativament, es compara la diferència que hi ha entre les components de la lletra i les components de cadascuna de les lletres del catàleg i finalment s'assigna la lletra amb la qual la diferència sigui menor. Tenint en compte que els valors i vectors singulars aporten informació diferent i que els vectors singulars tenen longitud variable en funció de la matriu de cada lletra s'ha definit un procediment per a calcular aquesta distància de la forma més exacta possible.

Per calcular aquesta distància, primer de tot es comparen les distàncies entre els valors singulars i entre els vectors singulars per separat. Com s'explica en 5.3, la informació aportada per els valors i vectors singulars decreix conforme s'agafen més valors. Per tant, només s'agafen els primers dos valors singulars i tres vectors singulars. Per aquest motiu s'aplica la següent ponderació a les distàncies obtingudes:

- Ponderació valors singulars:  $taxa\ vs = [1\ 0,4]$
- Ponderació vectors singulars de sortida:  $taxa\ u = [1\ 0,4\ 0,2]$
- Ponderació vectors singulars d'arribada:  $taxa\ v = [1\ 0,4\ 0,2]$

Per a obtenir aquestes ponderacions, primer de tot s'ha determinat de forma teòrica (5.3) nombre màxim de valors i vectors singulars necessaris per al reconeixement. Posteriorment, de forma experimental (8.4) s'han determinat els pesos d'aquests valors.

El càlcul de les distàncies és fa de la següent forma:

### 1. Comparació de valors singulars

Primer de tot es calcula la diferència entre els valors singulars de la lletra del text i la lletra del catàleg amb la qual s'està comparant, obtenint un vector de distàncies de longitud dos (dos valors singulars). Seguidament se li aplica la ponderació i es calcula la seva norma, obtenint així un únic valor que representa la distància entre els valors singulars: *distància valors singulars*.

$$valors\ singulars = valors\ singulars\ lletra - valors\ singulars\ catàleg$$

$$distància\ valors\ singulars = |taxa\ vs \cdot valors\ singulars|$$

on *taxa vs* és la ponderació dels valors singulars.

### 2. Comparació de vectors singulars

En segon lloc, es calculen les distàncies dels vectors singulars. A diferència dels valors singulars, cada vector singular no es un únic valor, sinó un vector de longitud variable per a cada lletra en funció de les dimensions de la seva matriu. Per això, s'ha definit un sistema per poder comparar vectors amb diferent longitud.

Aquest sistema consisteix en adaptar els dos vectors que s'estan comparant de forma que tinguin la mateixa longitud. Tenint un vector  $v$  (vector singular de la lletra) de longitud  $n$  i un vector  $c$  (vector singular del catàleg) de longitud  $m$ , cada valor del vector  $v$  quedarà repetit  $m$  vegades i cada valor del vector  $c$  quedarà repetit  $n$  vegades, quedant els dos vectors amb una longitud de  $n \cdot m$ .

$$v = (v_1\ v_2\ \dots\ v_n)$$

$$c = (c_1\ c_2\ \dots\ c_m)$$

$$v\ allargat = (v_{1,1}\ v_{1,2}\ \dots\ v_{1,m}\ v_{2,1}\ v_{2,2}\ \dots\ v_{2,m}\ \dots\ v_{m,1}\ v_{m,2}\ \dots\ v_{n,m})$$

$$c\ allargat = (c_{1,1}\ c_{1,2}\ \dots\ c_{1,n}\ c_{2,1}\ c_{2,2}\ \dots\ c_{2,n}\ \dots\ c_{m,1}\ c_{m,2}\ \dots\ c_{m,n})$$

Degut a aquesta adaptació, es necessari normalitzar els vectors singulars. Per tant, els vectors adaptats queden de la següent forma:

$$v_{adaptat} = \frac{v_{allargat}}{\sqrt{m}}$$

$$c_{adaptat} = \frac{c_{allargat}}{\sqrt{n}}$$

En el següent exemple podem veure aquest procediment:

$$v = (2 \ 3 \ 5) \cdot 0,1$$

$$b = (1 \ 2 \ 3 \ 4) \cdot 0,1$$

$$a_{adaptat} = \frac{(2 \ 2 \ 2 \ 2 \ 2 \ 3 \ 3 \ 3 \ 3 \ 5 \ 5 \ 5 \ 5) \cdot 0,1}{\sqrt{4}}$$

$$b_{adaptat} = \frac{(1 \ 1 \ 1 \ 2 \ 2 \ 2 \ 3 \ 3 \ 3 \ 4 \ 4 \ 4) \cdot 0,1}{\sqrt{3}}$$

Una vegada feta aquesta adaptació, els dos vectors singulars tenen la mateixa longitud i ja es pot calcular la diferència entre ells. Aquesta distància es calcula com la norma de la diferència entre els dos vectors singulars. Degut a les diferents orientacions de les lletres, els vectors podrien tenir també orientacions diferents. Per tant, es calcula tant la suma com la resta d'aquests dos vectors i s'agafa el resultat més petit. De forma que:

$$\text{distància vector singular} = \min(|(v \ ad) - (c \ ad)|, |(v \ ad) + (c \ ad)|)$$

on  $(v \ ad)$  és  $v_{adaptat}$  i  $(c \ ad)$  és  $c_{adaptat}$ .

Una vegada més, s'ha de normalitzar aquesta distància. Per tant:

$$\text{distància vector singular} = \frac{1}{\sqrt{n \cdot m}} \cdot \min(|(v \ ad) - (c \ ad)|, |(v \ ad) + (c \ ad)|)$$

Aquest procediment es realitza amb cada vector singular de sortida i d'arribada de la lletra, obtenint així dos vectors *distancia u* i *distancia v* que contenen les distàncies dels vectors singulars de sortida i d'arribada respectivament. Aplicant la ponderació i calculant les seves respectives normes s'obté un únic valor que representa la distància entre els vectors singulars: *distancia vectors singulars*.

$$\text{distància vectors singulars} = \sqrt{(\text{taxa } u)^2 \cdot (\text{distància } u)^2 + (\text{taxa } v)^2 \cdot (\text{distància } v)^2}$$

on *taxa u* i *taxa v* són les ponderacions dels vectors singulars de sortida i d'arribada respectivament i *distància u* i *distància v* són els vectors de distàncies dels vectors singulars de sortida i d'arribada respectivament.

### 3. Assignació final

Finalment, la distància global entre les components d'una lletra del text i una lletra del catàleg queda com:

$$\text{distància global} = \text{distància valors singulars} + \text{distància vectors singulars}$$

Cada lletra del text es compara amb totes les lletres del catàleg, quedant un vector amb les respectives distàncies. La lletra detectada és la corresponent a la distància més petita d'aquest vector de distàncies.

El procés finalitza quan s'han analitzat totes les lletres del text.

## 6.7. Determinació de l'èxit del reconeixement

Per tal de comprovar l'efectivitat d'aquest procés és necessari comparar els resultats obtinguts amb el text que hauria d'haver detectat. Amb aquest propòsit, en un fitxer de text es guarda el text real.

Una vegada realitzat el procés de reconeixement, es compara el resultat amb el text real determinant com a èxit si ha encertat la lletra i com errada si no s'ha encertat. Els resultats més significatius de la prova són: llargada del text (nombre de caràcters a detectar), nombre d'encerts i taxa d'encerts (percentatge d'èxits respecte a la llargada del text).

S'ha de tenir en compte que en el cas que el programa detecti més o menys lletres de les que hi ha al text en realitat (a causa de taques i altres imperfeccions) la detecció final queda desviada una posició i causa que la taxa d'encerts sigui significativament més baixa de la que és en realitat.

## 7. Implementació del codi

El reconeixement consta de tres funcions principals *segmentacio*, *signatures* i *OCR* que porten a terme les quatre fases del reconeixement.

A part de les funcions principals, el programa necessita altres funcions secundàries per el correcte funcionament del programa: *intervals\_zero*, *lectura*, i *resultats*.

L'script principal que controla les funcions anteriors s'anomena *correr\_fitxers*. Quan aquest s'executa, inicialitza les variables principals, executa la resta de funcions i retorna els resultats.

Per altra banda, es necessiten una sèrie de catàlegs amb els quals es comparen les components de les lletres. La funció *crear\_cataleg* crea aquests catàlegs a partir d'una plantilla.

Com s'explica a 6.2, les imatges s'han de tractar aplicant un flux de tinta. Aquest està compost per dos funcions: *aplicar\_flux* i *flux\_tinta\_negra*.

Finalment, per a realitzar l'estudi s'han utilitzat vàries funcions auxiliars que no intervenen en el codi principal.

### 7.1. Funcions principals

#### **segmentacio**

S'encarrega de dur a terme les dues primeres fases del procés: la binarització de la imatge i la segmentació per lletres.

Primer de tot transforma i guarda la imatge com una matriu de 1's i 0's. Seguidament detecta les línies que conté el text i elimina les anormalment petites (són una taca). Finalment detecta les lletres de cada línia, eliminant també les anormalment petites.

$$[lletres, linies, ll_linia] = segmentacio(imatge, format)$$

Variables d'entrada:

- *imatge*: una imatge en blanc i negre que conté lletres separades per espais en blanc i en línia recta
- *format*: el format de la imatge; en aquest cas TIFF.



Variables de sortida:

- *lletres*: matriu de quatre columnes i tantes files com lletres contingui la imatge, on els valors de les columnes són la fila inicial, fila final, columna inicial, columna final de cada lletra en aquest mateix ordre. Es a dir, la posició de les quatre cantonades de la regió que conté una lletra
- *linies*: matriu de dos columnes amb una fila per cada línia de text continguda en la imatge que indica la fila inicial i la fila final de cada línia
- *ll\_linia*: matriu de dos columnes amb una fila per cada línia de text que indica la paraula inicial i final de cada línia.

La matriu corresponent a la imatge queda guardada com a variable *A* al fitxer *imatgebinaria.mat* dins del mateix directori de treball de MATLAB.

### signatures

A partir de la matriu de cada lletra, crea una base de dades que conté la informació dels valors i vectors singulars de les lletres contingudes al text, portant a terme així la tercera fase del procés: obtenció de components.

$$[signatura] = signatures(lletres, dimensio, base)$$

Variables d'entrada:

- *lletres*: variable obtinguda a *segmentacio* que conté la informació de les posicions de les lletres a la imatge (convertida a matriu)
- *dimensio*: nombre de valors singulars que s'agafen de cada lletra
- *base*: nombre de vectors singulars de sortida i d'arribada que s'agafen de cada lletra.

Variable de sortida:

- *signatura*: estructura de tres columnes de cel·les i una fila per a cada lletra, on la primera columna és un vector que conté els valors singulars i la segona i tercera columna son matrius que contenen els vectors singulars de sortida i d'arribada respectivament.

Els valors singulars que s'obtenen es divideixen per el valor del primer i es guarden tants com indiqui la variable *dimensio* sense tenir en compte el primer (com s'explica a 0).

Respecte als vectors singulars de sortida i d'arribada, es guarden tants com la variable *base* indiqui.

Els valors de *dimensio* i *base* són 2 i 3 respectivament.

## OCR

Aquesta última funció s'encarrega de la última fase del reconeixement: la comparació.

Tal com s'explica detalladament a 6.6.2, es comparen les components de cada lletra amb cadascun dels caràcters continguts en un catàleg predefinit. D'aquesta forma, s'assigna com a solució final el caràcter del catàleg amb les components més semblants a les de la lletra que estem estudiant.

$$[deteccio\_numero, deteccio\_lletra] = OCR(signatura, nom\_cataleg, dimensio, base, taxa\_d, taxa\_u, taxa\_v)$$

Variables d'entrada:

- *signatura*: estructura amb les components de les lletres, obtinguda a la funció *signatures*
- *nom\_cataleg*: nom del catàleg amb el qual es volen comparar les lletres
- *dimensio*: nombre de valors singulars que s'agafen de cada lletra
- *base*: nombre de vectors singulars de sortida i d'arribada que s'agafen de cada lletra
- *taxa\_d*: vector amb els pesos corresponents als valors singulars
- *taxa\_u*: vector amb els pesos corresponents als vectors singulars de sortida
- *taxa\_v*: vector amb els pesos corresponents als vectors singulars d'arribada

Variables de sortida:

- *deteccio\_numero*: vector on cada posició conté l'índex del catàleg corresponent a la lletra detectada
- *deteccio\_lletra*: un cadena de text que conté les lletres detectades per el programa.

## 7.2. Funcions secundàries

### intervals\_zero

Subdivideix el vector donat en subinterval·s de píxels consecutius blancs o negres (es a dir, 1's o 0's) i retorna la posició d'aquests interval·s. La funció *segmentacio* crida aquesta funció per tal de detectar on comencen i acaben les línies de la imatge.

$$[nuls, nonuls] = intervals\_zero(v)$$

Variable d'entrada:

- *v*: vector de 1's i 0's.

Variables de sortida:

- *nuls*: posicions dels interval·s amb més de dos 0's consecutius
- *nonuls*: posicions dels interval·s amb més de dos 1's consecutius.

### lectura

Llegeix els caràcters continguts en un fitxer de text. S'utilitza per llegir el fitxer de text on es troben els caràcters reals del text i poder fer una comparació i obtenir una taxa d'encerts. També s'utilitza per a la creació de catàlegs.

$$[reallletres, realparaules, reallinees] = lectura(fitxer)$$

Variable d'entrada:

- *fitxer*: nom del fitxer que es vol llegir.

Variables de sortida:

- *reallletres*: cadena de text amb totes les lletres que conté el fitxer
- *realparaules*: matriu de dos columnes i una fila per cada paraula que conté el fitxer, que indica la posició inicial i final de cada paraula. Aquesta variable es purament informativa i no s'utilitza en aquest projecte, ja que totes les lletres del text a reconèixer estan separades per espais, independentment de si pertanyen o no a la mateixa paraula
- *reallinies*: matriu de dos columnes i una fila per cada línia que conté el fitxer, que indica la posició inicial i final de cada línia.

## resultats

Comparant *deteccio\_lletra* (de *OCR*) i *reallletres* (de *lectura*) es calcula el percentatge de lletres encertades. En un fitxer apart es guarden els següents els resultats: nom del catàleg utilitzat, llargada del text, nombre d'encerts, taxa d'encerts (percentatge), text detectat per l'OCR i text real. Addicionalment, en el cas que el programa hagués reconegut més o menys lletres de les reals (per exemple, a causa de taques), s'indicaria a aquest fitxer com "ERROR LLARGADA" .

*resultats (directori, nom, nom\_cataleg, deteccio\_lletra, reallletres)*

Variables d'entrada:

- *directori*: directori on es troba la imatge
- *nom* i *nom\_cataleg*: definides en el següent apartat
- *deteccio\_lletra* i *reallletres*: definides prèviament en l'apartat anterior

El fitxer que conté els resultats queda guardat amb el nom 'resultats\_' + *nom*. A la Figura 7.1 podem veure un exemple d'aquest fitxer.



Figura 7.1 Exemple d'un fitxer de text que conté els resultats d'un reconeixement

## 7.3. Control de les funcions

La funció **correr\_fitxers** controla tot el procés de reconeixement d'un text.

Les variables a modificar són:

- *nom*: nom dels fitxers del text que es vol reconèixer
- *nom\_cataleg*: nom del catàleg que es vol utilitzar per al reconeixement

Una vegada definides aquestes dues variables, es creen les variables: *taxa\_d*, *taxa\_u*, *taxa\_v*, *dimensio*, *base*, *directori*, *fitxer*, *format* i *imatge*. Finalment es criden les funcions en el següent ordre: *segmentacio*, *signatures*, *OCR*, *lectura* i *resultats*.

## 7.4. Creació dels catàlegs

La funció **crear\_cataleg** s'utilitza únicament quan es vol crear un nou catàleg a partir d'una plantilla que consta d'una imatge en blanc i negre que conté els caràcters manuscrits i un fitxer de text amb els caràcters corresponents.

Degut a que aquesta funció no es necessita per al procés de l'OCR, no té variables d'entrada ni de sortida. Per a crear un nou catàleg s'han de modificar les següents variables a l'script:

- *nom\_plantilla*: nom de la plantilla
- *nom\_cataleg*: nom amb el qual es vol guardar el catàleg.

Primer s'executen les funcions *segmentacio* i *signatures* per obtenir les components dels caràcters. Seguidament s'executa la funció *lectura* per obtenir *reallletres*, els caràcters reals corresponents a cadascuna de les signatures guardades. Finalment es guarden les següents variables en un fitxer ".mat":

- *alfa*: cadena de text que conté els caràcters de la plantilla
- *cataleg*: signatures dels caràcters de la plantilla

## 7.5. Funcions de flux de tinta

### flux\_tinta\_negra

Donada una imatge en blanc i negre en format TIFF, la sobreesciu aplicant el flux de tinta negra.

Com s'ha explicat a 6.2, cada punt de la matriu de 1's i 0's que representa la imatge, redueix la seva 'tinta' i traspasa part d'ella als píxels veïns. El procediment consisteix en reduir el valor de cada punt multiplicant-lo per una taxa (valor del 0 a l'1) i, a la vegada, sumar el valor d'aquest punt ponderat amb altra taxa (valor del 0 a l'1) als píxels veïns. El resultat final s'obté després de repetir aquest procediment un nombre de vegades.

*flux\_tinta\_negra (imatge\_flux,format,nfluxes,taxa\_cons,taxa\_veins)*

Variables d'entrada:

- *imatge\_flux*: nom de la imatge que es sobreescriu
- *format*: format de la imatge
- *nfluxes*: iteracions que realitza el programa de flux
- *taxa\_cons*: taxa de tinta que queda en un punt a cada iteració
- *taxa\_veins*: taxa de tinta que es traspassa als veïns superior, inferior i laterals

### **aplicar\_flux**

És la funció encarregada de portar a terme totes les accions necessàries per aplicar el flux a una imatge. No té variables d'entrada ni de sortida, només s'ha de modificar a l'script el nom de la imatge (*nom*) a la qual es vol aplicar el flux.

Aquesta funció crea una còpia de la imatge original amb el nom '*flux\_*' + *nom de la imatge* i després se li aplica el programa de flux a aquesta última. La resta de funcions utilitzaran aquesta imatge per a el reconeixement de forma automàtica.

## **7.6. Funcions auxiliars**

### **taxa\_encerts**

A l'hora de fer l'estudi iteratiu per trobar els valors òptims de *dimensio* i *base* així com els pesos més adequats per *taxa\_d*, *taxa\_u* i *taxa\_v* (com s'explica a 6.6.2), s'ha utilitzat aquesta funció per registrar les dades de totes les proves realitzades.

*taxa\_encerts* (*nom*, *deteccio\_lletra*, *reallletres*, *dimensio*, *base*, *taxa\_d*, *taxa\_u*, *taxa\_v*)

Les variables d'entrada s'han definit prèviament en diversos apartats d'aquest capítol.

Aquesta funció guarda en un fitxer tota la informació rellevant cada vegada que es realitza un reconeixement.

### **crear\_cataleg\_alfabet, crear\_cataleg\_alfanumeric i crear\_cataleg\_ABCD**

Degut a que les plantilles de l'alfabet i de nombres (on cada caràcter està repetit deu cops) no caben en una sola pàgina, s'han dividit en tres pàgines: majúscules, minúscules i

nombres. Una vegada creats els respectius catàlegs (*cataleg\_majuscules\_10*, *cataleg\_minuscules\_10* i *cataleg\_nombres\_10*), aquestes funcions creen un nou catàleg alfabètic o alfanumèric a partir dels mateixos.

## 7.7. Gestió documental

Per tal de treballar amb un cert ordre, la gestió documental s'ha realitzat de la següent forma:

### Estructura dels textos a reconèixer

Tots els arxius necessaris per al reconeixement d'un text o plantilla es troben a la mateixa carpeta amb al mateix nom. Cada text o plantilla necessita dos fitxers principals: la imatge original i un fitxer de text amb les lletres reals que conté la imatge. Quan s'aplica el programa de flux a la imatge original, es crea una de nova que queda guardada al mateix directori. Una vegada s'ha realitzat el procés de reconeixement de text, es desa un arxiu que conté els resultats, també en el mateix directori. A la Figura 7.2 podem veure un exemple del contingut d'una carpeta.

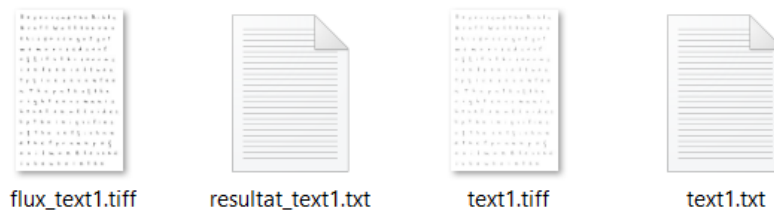


Figura 7.2 Contingut de la carpeta "text1"

### Estructura general

El directori on treballa MATLAB conté els següents arxius:

- totes les funcions esmentades en els apartats anteriors
- els catàlegs creats per la funció *crear\_cataleg*
- una carpeta "catàlegs" que conté les plantilles per a la creació dels catàlegs
- una carpeta "textos" que conté els textos per reconèixer amb el programa.

## 8. Estudis realitzats

Per tal d'obtenir els millors resultats possibles en el reconeixement de text s'han realitzat una sèrie d'estudis en els quals s'han determinat dos aspectes principals:

- Format del catàleg
- Pesos dels valors i vectors singulars

Tots els textos de l'estudi han sigut escrits per la mateixa persona i totes les imatges utilitzades han sigut pretractades de la forma següent:

1. Escaneig en escala de grisos amb una resolució de 600 ppp amb una impressora multifuncional *HP Deskjet F4500 series*.
2. Eliminació dels marges mitjançant el programa de la mateixa impressora *HP Scan and Capture*.
3. Transformació a format TIFF i monocromàtic (blanc i negre) mitjançant el programa *Gimp 2*.

Cal recordar que cada vegada que ens referim a un text o plantilla fem referència a la parella *imatge - fitxer de text*.

### 8.1. Catàlegs

A partir d'una sèrie de plantilles s'han creat els següents 7 catàlegs:

1. **cataleg\_alfabet:** conté les 26 lletres de l'alfabet anglès (majúscules i minúscules)
2. **cataleg\_nombres:** conté els nombres del 0 al 9
3. **cataleg\_alphanumeric:** és la combinació dels catàlegs *cataleg\_alfabet* i *cataleg\_nombres*
4. **cataleg\_alfabet\_10:** conté les 26 lletres de l'alfabet anglès (majúscules i minúscules) repetides cadascuna 10 cops
5. **cataleg\_nombres\_10:** conté els nombres del 0 al 9 repetits cadascun 10 cops
6. **cataleg\_alphanumeric\_10:** és la combinació dels catàlegs *cataleg\_alfabet\_10* i *cataleg\_nombres\_10*
7. **cataleg\_ABCD\_10:** extracte del *cataleg\_alfabet\_10* que conté les lletres A, B, C i D repetides cadascuna 10 cops.



## 8.2. Textos a estudiar

Per a fer totes les proves s'ha utilitzat un total de 8 textos:

1. **text1**: text de tipus alfabètic que conté tant majúscules con minúscules, amb una longitud de 213 caràcters
2. **text2**: continuació de *text1*, amb les seves mateixes característiques i una longitud de 180 caràcters
3. **text3**: continuació de *text2*, amb les seves mateixes característiques i una longitud de 104 caràcters
4. **ABCD**: conté les lletres A, B, C i D (com si es tractés de les respostes d'un examen tipus test) i una longitud de 77 caràcters
5. **nombres1**: només conté nombres, amb una longitud de 93 caràcters
6. **majuscles\_10**: conté totes les majúscules de l'alfabet anglès repetides cadascuna 10 cops, amb una longitud de 260 caràcters
7. **minuscules\_10**: conté totes les majúscules de l'alfabet anglès repetides cadascuna 10 cops, amb una longitud de 260 caràcters
8. **nombres\_10**: conté els nombres del 0 al 9 repetits cadascun 10 cops, amb una longitud de 100 caràcters.

## 8.3. Determinació del model del catàleg

En un primer moment es pot pensar que un catàleg que contingui una lletra amb una cal·ligrafia perfecta és suficient per tenir unes components de referència. En la realitat no és així, ja que intervenen factors com la posició de la pàgina on s'estigui escrivint (a dalt, a baix, lateral esquerre o dret...), la velocitat, la fatiga o inclús els canvis de cal·ligrafia a curt termini (mesos) que pot patir una persona.

És per això que s'han creat dos catàlegs: un amb una sola repetició per a cada caràcter i altre amb 10 repeticions per a cada caràcter. Cadascuna d'aquestes repeticions s'han escrit amb petites variacions, condicionades per els factors descrits anteriorment.

Per a poder arribar a la conclusió de quin del dos models de catàleg és més efectiu s'han realitzat les següents proves:

Nom	Model del catàleg	Catàleg	Taxa d'encerts
text1	1 repetició	catàleg_alfabet	67,2%
		catàleg_alfanumeric	67,1%
	10 repeticions	catàleg_alfabet_10	86,9%
		catàleg_alfanumeric_10	86,9%
nombres1	1 repetició	catàleg_nombres	58,1%
		catàleg_alfanumeric	58,1%
	10 repeticions	catàleg_nombres_10	96,8%
		catàleg_alfanumeric_10	92,5%

Taula 8.1 Proves realitzades amb els diferents models de catàleg

Com podem veure a Figura 8.1 s'observa una gran diferència entre els dos models de catàleg. Clarament el catàleg amb més taxa d'encerts és el de 10 repeticions.

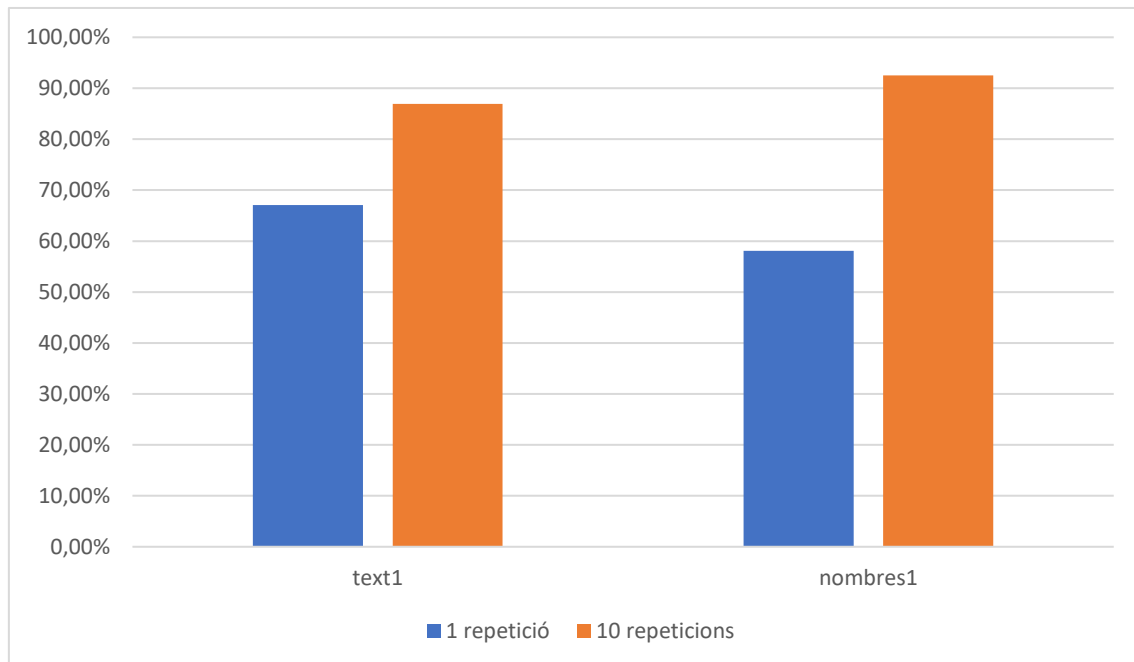


Figura 8.1 Gràfic amb els resultats obtinguts amb els dos models de catàleg

Per tant, a partir d'aquest punt de l'estudi, s'utilitzaran els següents catàlegs:

- Tipus alfabètic: *cataleg\_alfabet\_10*
- Tipus numèric: *cataleg\_nombres\_10*
- Tipus alfanumèric: *cataleg\_alphanumeric\_10*
- Tipus ABCD: *cataleg\_ABCD\_10*

Cal aclarir que els diversos estudis de tot el projecte han sigut molt iteratius i no s'han realitzat exactament en mateix ordre que es mostra en aquest capítol. En un primer moment no es tenien els valors definitius dels pesos dels valors i vectors singulars, pel que es va realitzar un estudi inicial per determinar quin dels dos models de catàleg era el més apropiat. Els resultats mostrats en aquest apartat són els corresponents a un estudi final de comprovació, realitzat amb els valors definitius dels pesos determinats en 8.4.

## 8.4. Determinació dels pesos dels valors i vectors singulars

Com s'ha explicat en 5.3 els valors i vectors singulars no aporten la mateixa informació. Igualment, la informació aportada per cada valor o vector singular va decreixent conforme s'agafen més. De forma que la informació aportada per el primer valor singular és molt més rellevant que la que aporten el quart, cinquè o sisè valor. A més, aquests últims poden induir a un error d'identificació. Per tant, és necessari ponderar la informació aportada per cada valor i vector singular.

L'objectiu d'aquest estudi és identificar quants valors i vectors singulars és necessiten i quines son les seves ponderacions per a obtenir un reconeixement òptim. Per tal que hi hagi el mateix nombre de repeticions de cada caràcter s'han estudiat els següents textos: *majuscules\_10*, *minuscules\_10* i *nombres\_10*. El catàleg utilitzat ha sigut de tipus alfanumèric. A la següent taula podem veure les proves que s'han realitzat:

Nom	Combinació	Pes valors singulars	Pes vectors singulars de sortida	Pes vectors singulars d'arribada	Taxa d'encerts
majúscules_10		[1 1 1]	0	0	18,1%
		0	[1 1 1]	[1 1 1]	78,8%
		0	[1 1 1]	0	70,8%
		0	0	[1 1 1]	54,2%

Nom	Combinació	Pes valors singulars	Pes vectors singulars de sortida	Pes vectors singulars d'arribada	Taxa d'encerts
	1	[1 1 1]	[1 1 1]	[1 1 1]	79,2%
	2	[1 0,5 0,2]	[1 0,5 0,2]	[1 0,5 0,2]	90%
	3	[1 1]	[1 1]	[1 1]	86,2%
	4	[1 0,5]	[1 0,5]	[1 0,5]	90%
	5	[1 0,5]	[1,5 0,5]	[1,5 0,5]	89,6%
	6	[1 0,5]	[1 0,5 0,2]	[1 0,5 0,2]	90%
	7	[1 0,4]	[1 0,4 0,2]	[1 0,4 0,2]	90,4%
minúscules_10	1	[1 1 1]	[1 1 1]	[1 1 1]	94,2%
	2	[1 0,5 0,2]	[1 0,5 0,2]	[1 0,5 0,2]	96,5%
	3	[1 1]	[1 1]	[1 1]	96,2%
	4	[1 0,5]	[1 0,5]	[1 0,5]	96,5%
	5	[1 0,5]	[1,5 0,5]	[1,5 0,5]	95,4%
	6	[1 0,5]	[1 0,5 0,2]	[1 0,5 0,2]	96,5%
	7	[1 0,4]	[1 0,4 0,2]	[1 0,4 0,2]	95,8%
nombres_10	1	[1 1 1]	[1 1 1]	[1 1 1]	88%
	2	[1 0,5 0,2]	[1 0,5 0,2]	[1 0,5 0,2]	89%
	3	[1 1]	[1 1]	[1 1]	84%
	4	[1 0,5]	[1 0,5]	[1 0,5]	87%
	5	[1 0,5]	[1,5 0,5]	[1,5 0,5]	87%
	6	[1 0,5]	[1 0,5 0,2]	[1 0,5 0,2]	89%
	7	[1 0,4]	[1 0,4 0,2]	[1 0,4 0,2]	91%

Taula 8.2 Proves realitzades per a determinar els pesos dels valors i vectors singulars

Com podem veure a la Taula 8.2, primer s'han realitzat 4 proves per veure com afecten de forma individual els valors i vectors singulars. S'ha pogut determinar que els vectors singulars són més importants que els valors singulars. Fet això, s'han analitzat els 3 textos amb 7 combinacions de diferents de pesos. Podem observar aquests resultats de forma més visual a la Figura 8.2.

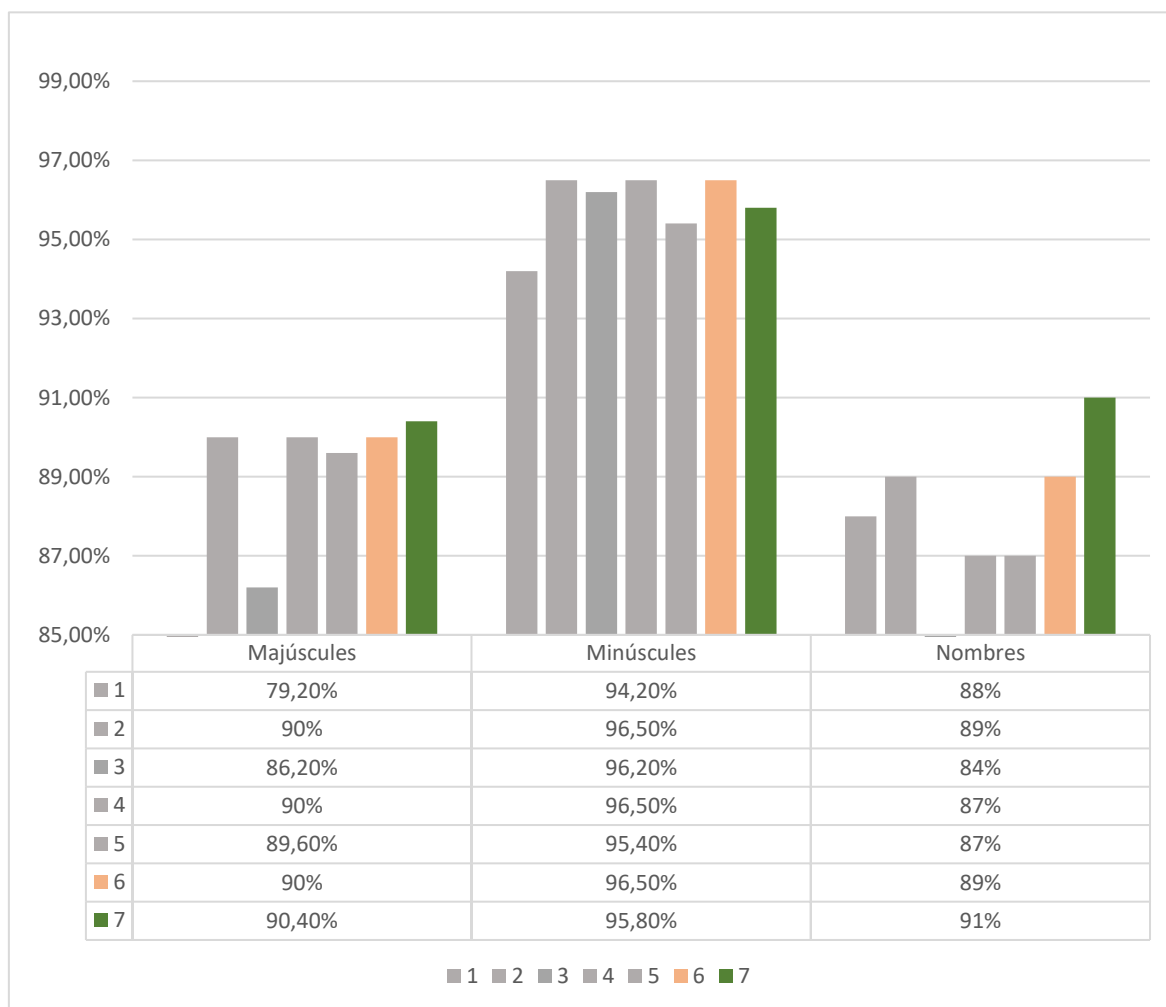


Figura 8.2 Comparació dels resultats obtinguts amb diferents combinacions de pesos

Observem que tant en el cas de les majúscules com dels nombres, la taxa d'encerts més elevada s'obté amb els següents pesos (combinació 7):

- Pesos valors singulars: [1 0,4]
- Pesos vectors singulars (de sortida i d'arribada): [1 0,4 0,2]

En el cas de les minúscules la taxa més elevada s'ha aconseguit amb la combinació 6. L'empitjorament de la taxa obtinguda amb la combinació 7 queda compensada amb la millora de la taxa de les majúscules i els nombres. Per tant, aquests seran els pesos definitius.

## 8.5. Anàlisi d'errors típics

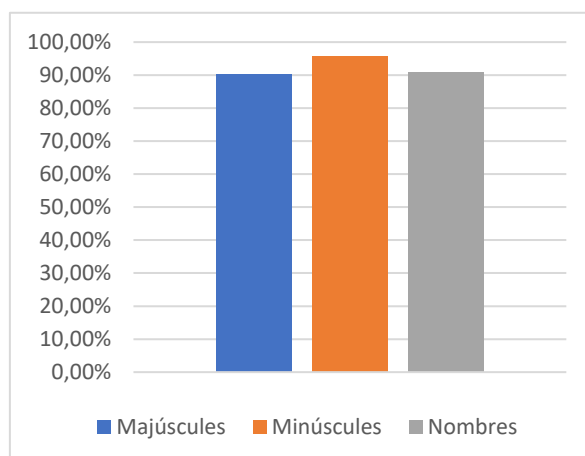
Un cop ja tenim definits el model de catàleg i els pesos dels valors i vectors singulars s'ha realitzat un estudi per identificar quins són els errors típic d'identificació.

### 8.5.1. Discriminació per tipus de caràcter

Primer de tot, s'ha volgut diferenciar entre majúscules, minúscules i nombres.

Nom	Taxa encerts
majuscles_10	90,4%
minuscles_10	95,8%
nombres_10	91%

*Taula 8.3 Taxa d'encerts discriminant per majúscules, minúscules i nombres*



*Figura 8.3 Gràfic dels resultats obtinguts discriminant per majúscules, minúscules i nombres*

Com es pot comprovar a la Figura 8.3 les minúscules són les que obtenen una major taxa de reconeixement i les majúscules i els nombres obtenen un reconeixement semblant.

### 8.5.2. Parelles de caràcters

Degut a la similitud que hi ha entre algunes lletres i també entre algunes lletres i nombres és possible detectar erròniament una lletra i confondre-la amb una semblant. Amb la finalitat de detectar les parelles de caràcters que es confonen habitualment, s'han analitzat *text1*, *text2*, *text3*, *nombres1*, *ABCD*, *majúscules\_10*, *minúscules\_10* i *nombres\_10* i s'han identificat els casos més recurrents:

Lletres		Nombre d'aparicions	Nombre d'identificacions errònies	Taxa d'error d'identificació
t	l	85	14	16,5%
M	N	20	3	15%
y	Y	34	5	14,7%
5	E	28	4	14,3%
n	k	25	3	12,0%
0	O	27	3	11,1%
z	Z	22	2	9,1%
9	q	26	2	7,7%
3	S	31	2	6,5%
w	W	33	2	6,1%
u	v	37	2	5,4%
D	R	40	2	5%
t	r	89	4	4,5%
a	k	50	2	4,0%
f	l	55	2	3,6%
z	s	57	2	3,5%
b	h	60	2	3,3%
6	S	31	1	3,2%

Taula 8.4 Identificacions errònies de parelles de caràcters

Les dos primeres columnes de la Taula 8.4 mostren els caràcters que es confonen més habitualment degut a la seva similitud. Podem observar gràficament a Figura 8.4 quina és la freqüència amb la que s'acostumen a cometre aquests errors quan es vol detectar una de les dos lletres.

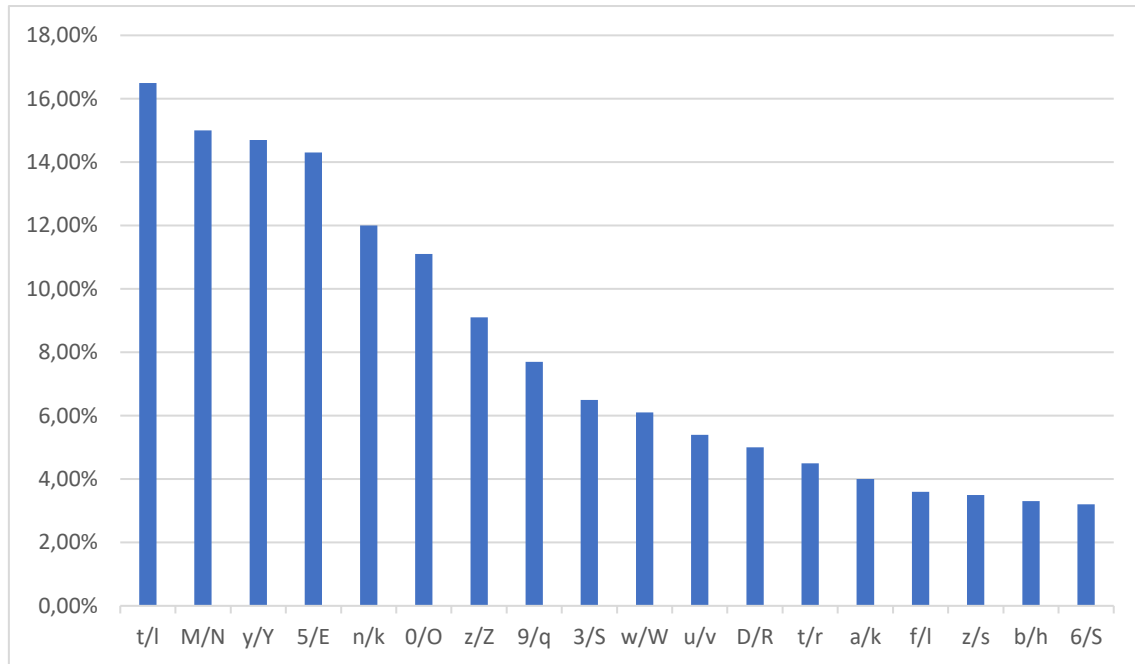


Figura 8.4 Taxa d'error d'identificació de les parelles de lletres que es confonen habitualment

Cal aclarir que alguns d'aquests errors són causats per la mateixa persona que escriu el text, ja que en alguns casos, quan s'escriu ràpid, algunes lletres diferents s'acostumen a escriure de forma molt similar. Exemples d'això poden ser els casos de y/Y, w/W i u/v.

## 8.6. Estudis amb el sistema final

Finalment, s'ha volgut comprovar l'efectivitat del sistema aplicant l'algoritme a una sèrie de casos pràctics:

Nom	Tipus de catàleg	Llargada	Encerts	Taxa encerts
text1	Alfabètic	213	185	86,9%
	Alfanumèric		185	86,9%
text2	Alfabètic	180	168	93,3%
	Alfanumèric		167	92,8%
text3	Alfabètic	104	95	91,3%
	Alfanumèric		95	91,3%



Nom	Tipus de catàleg	Llargada	Encerts	Taxa encerts
ABCD	ABCD	77	74	96,1%
nombres1	Numèric	93	90	96,8%
	Alfanumèric		86	92,5%

Taula 8.5 Taxa d'encert d'alguns casos pràctics

Les tres primeres proves *text1*, *text2* i *text3* formen part d'un mateix text que s'ha dividit en tres pàgines. La taxa conjunta dels tres textos és d'un 90,1%. Veiem també que, en contra del que es podia pensar abans de fer les proves, la taxa d'encerts és la mateixa (excepte la diferència d'un error a *text2*) independentment de si s'utilitza els catàleg alfabètic o alfanumèric.

Amb el text *ABCD*, com el catàleg consta només d'aquestes 4 lletres, la taxa augmenta fins a un 96,1%. Els 3 errors comesos han sigut el mateix: la D s'ha reconegut com una B.

En el cas des *nombres1* s'arriba a una taxa de 92,5% amb el catàleg alfanumèric. Coneixent que el text només conté nombres es pot augmentar aquesta taxa fins al 96,8% utilitzant el catàleg numèric.

Arribat a aquest punt, es dona per finalitzat l'estudi.

## 9. Planificació i programació

Aquest projecte ha estat realitzat des d'octubre de 2016 fins a gener de 2017. Durant aquest temps s'han abordat diferents fases i realitzat diverses tasques.

	2016												2017
	Octubre				Novembre				Desembre				Gener
Documentació													
Desenvolupament													
Estudis													
Redacció													

Figura 9.1 Diagrama de planificació del projecte

Com podem veure a Figura 9.1 primer es va realitzar una fase de documentació en la que es va obtenir la informació necessària per a realitzar aquest projecte. Aquesta fase de documentació ha sigut recurrent durant tot el temps que ha durat el projecte.

Després es va portar a terme la fase de desenvolupament de l'algoritme a MATLAB. Una vegada es va arribar al 80% de desenvolupament es van a començar a fer estudis, els resultats dels quals van ajudar a acabar de definir la fase de desenvolupament. Finalment, amb la realització dels últims estudis ja es va a començar a redactar la memòria del projecte.

A part d'aquestes fases, durant el termini del projecte també s'han realitzat reunions de seguiment de forma periòdica, a les quals han assistit l'autor i el director de projecte.

A la Taula 9.1 podem veure el temps de dedicació a cadascuna de les tasques.

	<i>Documentació</i>	<i>Desenvolupament</i>	<i>Estudis</i>	<i>Redacció</i>	<i>Reunions</i>	<b>Total</b>
Hores	40	150	50	90	10	<b>340</b>
% Dedicació	12%	44%	15%	26%	3%	<b>100%</b>

Taula 9.1 Dedicació de cada tasca

## 10. Pressupost

Aquest projecte ha estat en la major part desenvolupament informàtic. Les despeses associades consten del material que s'ha utilitzat per a la seva execució, material per a la realització dels textos manuscrits i la matrícula universitària. A la Taula 10.1 podem veure aquestes despeses en detall:

Recurs	Preu
Ordinador Toshiba Satellite	1000 €
Impressora multifuncional HP Deskjet F4500 series	120 €
Llicència Office	80 €
Pendrive Sandisk 32 Gb	20 €
Material d'escriptura i fulls	10 €
Matrícula universitària TFG	300€
<b>Total</b>	<b>1530 €</b>

*Taula 10.1 Despeses*

Cal aclarir que, excepte la matrícula, ja es disposava de tots aquests recursos quan es va començar el projecte. A més, la llicència de MATLAB té un cost de 2000 €, però s'ha utilitzat la versió d'estudiant gratuïta proporcionada per la UPC.

## **11. Impacte sobre l'entorn**

### **11.1. Impacte mediambiental**

Com ja s'ha mencionat, aquest es un projecte principalment de desenvolupament informàtic. Per tant, té un impacte ambiental pràcticament negligible.

Tots els fulls utilitzats durant aquest projecte han sigut reciclats. De forma que l'impacte ambiental causat per aquest projecte ha sigut aquell derivat de la producció des recursos materials que s'han utilitzat i de la producció d'electricitat que s'ha fet servir per el funcionament de l'ordinador.

### **11.2. Impacte social**

Avui en dia tot tendeix a digitalitzar-se. En el cas dels documents, la seva digitalització té una sèrie de beneficis: estalvi d'espai físic, més accessibilitat i control de la informació, més dificultat de perdre-la. Moltes institucions dediquen molt de temps i recursos en traspasar documents manuscrits a format digital. La introducció d'un sistema de reconeixement de text manuscrit estalviaria moltes hores de treball.

Per tant, aquest projecte afecta positivament en la societat en general, en tant a que aporta un avanç en la investigació de les tecnologies de reconeixement de text.

## Conclusions

Gràcies al projecte previ a aquest (Berguedà, 2016) ja es sabia que per reconèixer text tipografiat eren necessaris els 5 primers valors singulars i els primers vectors singulars de sortida i d'arribada (per poder diferenciar lletres simètriques). En aquest treball, en canvi, es va plantejar la hipòtesi de que es necessitaria donar més importància als vectors singulars per poder detectar text manuscrit. Després de realitzar tots els estudis s'ha pogut comprovar que aquests no només tenen un paper fonamental per al reconeixement de text manuscrit, sinó que són inclús més importants que els valors singulars.

Un altre punt a determinar ha sigut com tractar les dades per tal que aquestes siguin invariants als canvis d'escala de la imatge. Aquest problema s'ha solucionat obtenint els valors singulars com a quocients de cada valor singular per el valor del primer. Cal recordar que, degut a això, s'ha prescindit del primer valor singular ja que el seu quocient és 1.

També s'ha determinat que la informació que aporten els valors i vectors singulars es decreixent en funció de la seva posició. Per tant, s'ha plantejat la possibilitat de donar més o menys importància a cada valor o vector singular segons la seva posició. Efectivament, la precisió ha augmentat quan s'ha donat més importància al primer valor i vector, i disminuint aquesta importància segons la seva posició. Tenint tot això en compte, s'ha arribat a la conclusió que el nombre de valors i vectors singulars òptims per al reconeixement són 2 i 3 amb uns pesos de  $[1 \ 0,4]$  i  $[1 \ 0,4 \ 0,2]$  respectivament.

Respecte al tractament de les imatges han sigut molt importants el programa de flux de tinta i la eliminació de taques a la fase de segmentació. S'han pogut reforçar les lletres i eliminar les taques de forma totalment efectiva, evitant així que es produeixin errors de reconeixement d'aquest tipus.

Un cop desenvolupat tot el codi, s'han fet una sèrie de proves i s'han obtingut uns resultats que oscil·len entre el 87% i el 97%. Aquest és un percentatge molt elevat considerant que es tracta de text manuscrit. Al principi del projecte s'estimava que no s'arribaria al 20% d'encerts. Per tant podem afirmar que en aquest sentit el projecte ha sigut un èxit.

S'ha de tenir en compte que les proves s'han realitzat amb dues restriccions sense les quals no s'hauria pogut realitzar el reconeixement: text recte i lletres separades per espais (encara que es tracti de la mateixa paraula). Aquesta última condició és la més restrictiva, ja que no permet que es reconeguin paraules. També s'ha utilitzat un catàleg escrit per la mateixa persona que ha escrit els textos a reconèixer, pel que es pot deduir que si s'analitza un text escrit per altra persona amb altra cal·ligrafia la taxa d'encert es reduirà.

En general els objectius del projecte s'han complert amb resultats realment molt satisfactoris, amb taxes d'encert més elevades del que s'esperava en un principi. Encara que aquest sistema no pot diferenciar paraules ni reconèixer lletra lligada, la investigació portada a

terme indica que en un futur es podrà continuar explorant per tal de fer això una realitat.

## Agraïments

Principalment m'agradaria agrair a Jaume Amorós, tutor del projecte, per el suport donat durant tota la elaboració d'aquest treball. Sense els seus coneixements i la seva orientació al llarg d'aquests mesos hauria sigut impossible la realització d'aquest projecte.

També voldria agrair a la Judith Berguedà per el seu magnífic treball realitzat en el projecte previ a aquest (Berguedà, 2016). Els coneixements aportats pel seu projecte han sigut claus per a poder continuar en la investigació del reconeixement de text mitjançant la SVD.

## Bibliografia

- Amorós, J. (2010). Apunts de l'assignatura de geometria cursada en els graus d'enginyeria en tecnologies industrials, enginyeria química i enginyeria de materials. Barcelona.
- Berguedà, J. (2016). *Reconeixemen de text mitjançant la SVD*. Barcelona.
- Convertir imagen al formato TIFF*. (2016). Recollit de <http://imagen.online-convert.com/es/convertir-a-tiff>
- Horn, R., & Johnson, C. (1991). Topics in Matrix Analysis. *Cambridge University Press*, Thm 4.2.15.
- Kharma, & Ward. (1999). Recognition Systems for the Non-Expert. A *IEEE Canadian Review Autum* (p. 5-8).
- Meza, V. (2013). *Athento*. Recollit de <http://blog.athento.com/2013/11/reconocimiento-de-letra-manuscrita-estado-del-arte-de-las-tecnologias-icr.html>
- Mori, Suen, & Yamamoto. (1992). Historical Review of OCR Research and Development. A *Proceedings of the IEEE* (p. 1029-1058).
- Plamondon, & Srihari. (2000). On-Line and Off-Line Handwriting Recognition. A *IEEE Transactions on PAttern Analysis and Machine Intelligence* (p. 63-84).
- The MathWorks Inc. (2016). *MATLAB R2016b*. Recollit de <https://es.mathworks.com/help/matlab/>. Massachusetts
- Wikipedia*. (2016). Recollit de [https://es.wikipedia.org/wiki/Descomposici%C3%B3n\\_en\\_valores\\_singulares](https://es.wikipedia.org/wiki/Descomposici%C3%B3n_en_valores_singulares)
- Wikipedia*. (2016). Recollit de [https://es.wikipedia.org/wiki/Reconocimiento\\_%C3%B3ptico\\_de\\_caracteres#Binarizaci.C3.B3n](https://es.wikipedia.org/wiki/Reconocimiento_%C3%B3ptico_de_caracteres#Binarizaci.C3.B3n)
- Wikipedia*. (2016). Recollit de [https://es.wikipedia.org/wiki/Reconocimiento\\_%C3%B3ptico\\_de\\_caracteres](https://es.wikipedia.org/wiki/Reconocimiento_%C3%B3ptico_de_caracteres)
- Worldpress*. (9 / 11 / 2009). Recollit de <https://gupost.wordpress.com/2009/11/09/ocr-icr-y-omr-%C2%BFque-son-y-para-que-sirven/>
- Zaballa, I. (2012). *Valores singulares. ¿Qué son? ¿Para qué sirven?* Leioa.